Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A learning-based method to detect and segment text from scene images*

JIANG Ren-jie†, QI Fei-hu, XU Li, WU Guo-rong, ZHU Kai-hua

(*Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai 200240, China*)

†E-mail: blizard1982@sjtu.edu.cn

Received Jun. 1, 2006; revision accepted Oct. 10, 2006

Abstract: This paper proposes a learning-based method for text detection and text segmentation in natural scene images. First, the input image is decomposed into multiple connected-components (CCs) by Niblack clustering algorithm. Then all the CCs including text CCs and non-text CCs are verified on their text features by a 2-stage classification module, where most non-text CCs are discarded by an attentional cascade classifier and remaining CCs are further verified by an SVM. All the accepted CCs are output to result in text only binary image. Experiments with many images in different scenes showed satisfactory performance of our proposed method.

Key words: Text detection, Text segmentation, Text feature, Attentional cascade
doi:10.1631/jzus.2007.A0568 **Document code:** A **CLC number:** TP391.41

INTRODUCTION

Text is ubiquitous in our daily life, on road signs, bill boards, shop names, menus, labels and so on. Extracting and recognizing text has a promising future in information retrieval, auto-driving system, assistance of visually impaired people or travellers abroad. As OCR (Optimal Character Recognition) can only deal with text in simple background, text extraction is the preliminary procedure for recognition in variant scenes. If we can detect and segment text from natural scene images, it will be very helpful for many important applications.

Approaches of many studies conducted to this field can be divided into two categories: region-based and texture-based.

Region-based methods use the properties of the color or gray scale in a text region or their differences with the corresponding properties of the background. These methods can be further divided into two groups: connected component (CC) based and edge based. These two approaches work in a bottom-up fashion: first, they identify sub-structures in pictures, such as CCs (Wang and Kangas, 2003; Zhang and Chang, 2004) or edges (Kim *et al*., 2004; Liu C. *et al*., 2005; Lyu *et al*., 2005; Takahashi and Nakajima, 2005), and then merge these sub-structures to mark bounding boxes for text by heuristic rules (Wang and Kangas, 2003; Lyu *et al*., 2005) or learning-based rules such as neural networks, Markov Random Field (Zhang and Chang, 2004), etc.

Texture-based methods use the observation that text in images has distinct textural properties that distinguish it from the background. The techniques based on Fast Fourier Transform (Chun *et al*., 1999), Discrete Cosine Transform (Qian and Liu, 2006), Gabor (Chen *et al*., 2001; Liu C.L. *et al*., 2005), wavelet (Mao *et al*., 2002), spatial variance (Clark and Mirmehdi, 2000; Kim *et al*., 2003; Ekin, 2006), etc. can be used to detect the textural properties of text regions in an image. In the stage of verification, heuristic rules, neural network (Chun *et al*., 1999), Support Vector Machine (SVM, Kim *et al*., 2003), con-

* Project supported by the OMRON and SJTU Collaborative Foundation under PVS project (2005.03~2005.10)

ditional random field (Weinman *et al*., 2004) are popular.

In this paper, we propose a novel CC-based algorithm. First, the input image is decomposed into multiple CCs by Niblack clustering algorithm including text CCs and non-text CCs. Then, all of the CCs are input into the classification module. To segment text from background, our purpose is to eliminate non-text CCs but preserve text CCs. The most significant difference between our method and that of others is that we build a 2-stage classification module including a coarse classification stage and a refined classification stage. The former is implemented by a cascade classifier, in which most apparent non-text CCs are discarded as early as possible to save a great deal of computation. The latter is implemented by an SVM concentrating on CCs accepted by the cascade and doing further verification. Only those accepted by both cascade classifier and SVM are output in the final result. Both of the two classifiers are based on machine learning. The combination of weak classifier and strong classifier guarantees the effectiveness and efficiency of our approach.

IMAGE DECOMPOSITION

The quality of CCs is critical to the performance of a CC-based method. In natural scenes, due to illumination variation and noise, a robust clustering approach is required in the situation. We employ Niblack (Winger *et al*., 2000) to decompose original image into CCs (Eq.(1)).

$$Niblack(x,y)=\begin{cases}1, & f(x,y)>T_{+}(x,y),\\ -1, & f(x,y)<T_{-}(x,y),\\ 0, & \text{otherwise},\end{cases} \tag{1}$$

$$T_{\pm}(x,y)=\mu(x,y,W)\pm k\cdot\sigma(x,y,W), \tag{2}$$

where $f(x,y)$ is intensity of the original image at (x,y), W is the size of sub-window centered at (x,y), $\mu(x,y,W)$ and $\sigma(x,y,W)$ are intensity mean and deviation of sub-window respectively, k is predetermined constant.

We apply Niblack algorithm on input image (W=20×20, k=0.185) and get resulting image with 3 color layers: black (−1), white (1) and gray (0). Both black and white layers are foreground containing text while gray layer is background and ignored. CCs in black and white layers compose the candidate CCs of the input image including text and non-text. Later, they will be fed into a 2-stage verification module.

In this paper, each CC keeps its central color and the map of pixels belonging to it. Central color (r_c, g_c, b_c) is the average color of all pixels in the CC:

$$(r_c,g_c,b_c)=\frac{1}{n}\sum_{i=1}^{n}(r_i,g_i,b_i), \tag{3}$$

where n is the number of pixels belonging to the CC, (r_i, g_i, b_i) is color of each pixel in the CC.

Fig.1 shows the result of decomposition. The original image (Fig.1a) is clustered by Niblack (Fig.1b), and Fig.1c is the map of all CCs.

(a) (b) (c)

Fig.1 Result of Niblack: (a) Original image; (b) Niblack image; (c) Connected component map

TEXT FEATURES

As we know, success of classification depends on the quality of features. Therefore, the selection of text features is important for the following classification.

Totally 17 features were developed to discriminate text CCs from non-text CCs in our method (Table 1). All these features can be divided into 5 categories: geometric, shape regularity, edge, stroke, and spatial coherence.

Geometric features are used to measure the basic information on CCs, which are used to discard the most apparent non-text CCs. This group contains *AreaRatio*, *LengthRatioL*, *LengthRatioS*, *AspectRatio*, *DotsRatio*, *BorderDist_X*, *BorderDist_Y*. Where, *CC* is the input Connected-Component, *w* and *h* are width and height of *CC* respectively, (*tlx*, *tly*) and (*brx*, *bry*) are top-left and bottom-right point of *CC* respectively, |·| counts the pixels belonging to the *CC*, *area*(·) cal-

culates the area of input *CC*, *PicW* and *PicH* are picture's width and height respectively.

Table 1 Text features

Feature	Meaning
AreaRatio	$area(CC)/area(Pic)$
LengthRatioL	$\max(w/PicW, h/PicH)$
LengthRatioS	$\min(w/PicW, h/PicH)$
AspectRatio	$\min(w/h, h/w)$
DotsRatio	$\lvert CC\rvert/area(Pic)$
BorderDist_X	$\min(tlx, PicW-brx)$
BorderDist_Y	$\min(tly, PicH-bry)$
HoleNum	$imholes(CC)$
OccupyRatio	$\lvert CC\rvert/area(CC)$
Compactness	$area(CC)/\lvert contour(CC)\rvert^2$
ContourRoughness	$\lvert CC-open(CC,2\times 2)\rvert/\lvert CC\rvert$
EdgeContrast	$[Border(CC)\cap Edge(Pic)]/Border(CC)$
EdgeAngleSym	$\sum_{\theta=0}^{\pi}\lvert A(\theta)-A(\theta+\pi)\rvert$
StrokeWidthMean	$Mean\{strokeWidth[skeleton(CC)]\}$
StrokeWidthDev	$\dfrac{Deviation\{strokeWidth[skeleton(CC)]\}}{Mean\{strokeWidth[skeleton(CC)]\}}$
BackgroundInfo	$area[background(CC)]/area(Pic)$
Cohesion	$area[dilate(CC,5\times 5)]/area(Pic)$

Shape regularity features measure the shape regularity for further exploiting the difference between texts and non-texts such as *HoleNum*, *OccupyRatio*, *ContourRoughness* and *Compactness*. Where, *imholes*(·) is morphological hole counting, *contour*(·) gets all contour pixels of the input CC, *open*(·, *strel*) is morphologically open with structure element *strel*.

Edge features are the intrinsic features of characters. Two edge features are used in this paper: *EdgeContrast* and *EdgeAngleSym*. *EdgeContrast* measures the contrast between characters and the background, where *Edge*(·) is Canny edge map of the input image. By statistics on text CCs' edge angle (Clark and Mirmehdi, 2000), we find the symmetry of contour pixels' angle distribution. While for non-text CCs, this symmetry does not exist. Then we use feature *EdgeAngleSym* to measure CCs' angle symmetry, where $A(\theta)$ is the number of CC's contour pixels whose angles are θ.

Stroke features also reveal the essence of text. Usually, a character is composed of strokes with small and uniform width. Two features *StrokeWidthMean* and *StrokeWidthDev* exploit the 'smallness' and 'uniformity' respectively, where *skeleton*(·) is morphological skeleton operation and *strokeWidth*(·) stands for the shortest distance between one pixel on skeleton to the outside pixels.

All features described above are based on a single CC. However, other features can be extracted from the spatial relationship between multi-CCs for improving the performance of classification. Such kinds of features are *BackgroundInfo* and *Cohesion*, where *background*(·) is the smallest CC which contains the input CC and *dilate*(·, *strel*) is morphological dilation with structure element *strel*.

COARSE CLASSIFICATION

Cascade structure

In the stage of coarse classification, a cascade classifier is employed to discard apparent non-text CCs (Zhu *et al*., 2005). The cascade classifier consists of a series of weak classifiers, each concentrating on one feature mentioned in the section "TEXT FEATURES". Fig.2 illustrates the structure of cascade.

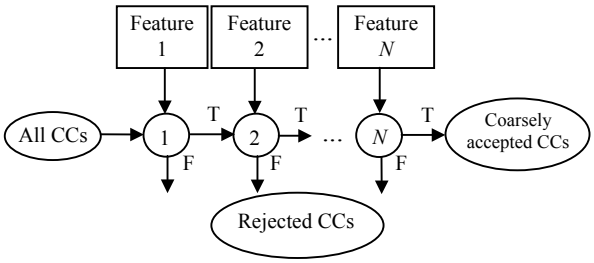


Fig.2 Structure of the cascade classifier

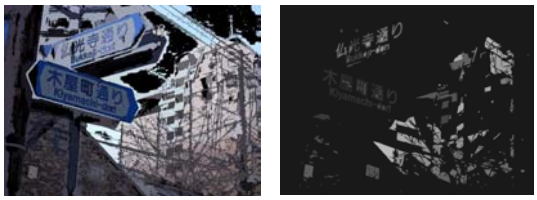
In this paper, a weak classifier is composed of a feature and two thresholds: one upper threshold and one lower threshold (Eq.(4)). For each input CC, the weak classifier measures the feature and makes the decision whether the CC is text or not.

$$h_i(x)=\begin{cases}1, & \text{if } \theta L_i < f_i(x) < \theta U_i,\\ 0, & \text{otherwise,}\end{cases} \tag{4}$$

where x is the input CC, f_i is the feature of the ith weak classifier, θL_i and θU_i are the lower and upper threshold of the ith weak classifier respectively, h_i is

the decision of the ith weak classifier. If h_i=1, the CC is regarded as text. Otherwise, the CC is considered as non-text.

At the beginning, all CCs extracted in the decomposition step are fed into the first weak classifier. It measures certain feature on CCs one by one and categorizes them into positive or negative group. The negative CCs are considered as non-text and rejected immediately. For positive CCs, similar processing is repeated in the following weak classifiers until the end of the cascade. When this cascade is finished, about 90% non-text CCs are discarded and almost all the text CCs are preserved. Fig.3 demonstrates the result of the cascade.



(a) (b)

Fig.3 Effect of the cascade classifier. (a) Input: all extracted CCs; (b) Output: CCs accepted by the cascade

What is more important, all features used by SVM in refined classification stage are prepared in the cascade. Though SVM has satisfactory discriminating ability, its tolerance of feature absence is poor. So we must calculate all features for all input CCs before SVM can classify them. Without the cascade, the system would be quite computationally exhaustive. Due to the advantage of cascade, there is no need to calculate all the 17 features for all CCs. We compile statistics on 500 testing images and the advantage of cascade is shown in Fig.4. Number of non-text CCs decreases after each round of weak classification. Since most non-texts are rejected in the early stage of cascade, a great deal of meaningless computation is reduced.

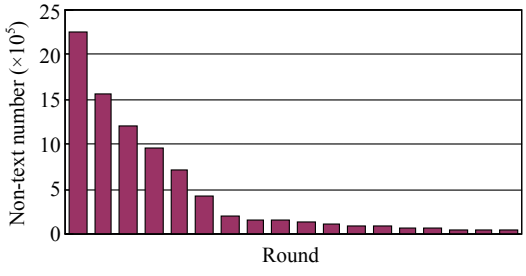


Fig.4 Cascade process for non-text CCs

Cascade training

Since we know the structure and advantage of the cascade classifier, the remaining problem is how to train it. This subsection discusses the details of cascade training. Before we go to the training process, we will clarify some notations (Table 2).

Table 2 Notations for the cascade training

Notation	Meaning
positive	All of the text CCs
negative	All of the non-text CCs
hit	Accepted text CCs
error	False-accepted non-text CCs
f	False-positive rate: $\frac{area(error)}{area(negative)}$
d	Detection rate: $\frac{area(hit)}{area(positive)}$
P	Positive training set
N_i	Negative training set in the ith layer
f_i	Maximum false-positive rate in the ith layer
d_i	Minimum detection rate in the ith layer
F	Overall maximum false-positive rate
D	Overall minimum detection rate
M	Number of weak classifiers
h_i	The ith weak classifier in the cascade

From the observation that the cascade classifier consists of 17 weak classifiers and that each will reject non-text CCs immediately, the overall detection rate D and false-alarm rate F have a close relationship with d_i and f_i respectively of each weak classifier. The relationship is shown in Eq.(5):

$$D=\prod_{i=1}^{M} d_i, \qquad F=\prod_{i=1}^{M} f_i,$$
$$\log D=\sum_{i=1}^{M}\log d_i, \quad \log F=\sum_{i=1}^{M}\log f_i. \tag{5}$$

In the logarithm conversion of the basic relationship, we will find that the overall detection rate D is linearly dispatched into separate weak classifiers. Given the minimum detection rate d_i, one weak classifier can decide its θL_i and θU_i easily. Now, the problem is how to dispatch the overall detection rate.

We make the assumption that detection rate is dispatched according to the 'quality' of each weak classifier. In Eq.(6), D_{dispatch} is target detection rate which can be dispatched and γ_i is 'quality' portion of

each weak classifier.

$$d_i = (D_{\text{dispatch}})^{\gamma_i}. \quad (6)$$

Fig.5 shows the details on the training process. In the training, we regard false-alarm rate f_i as 'quality' portion of each weak classifier. Note that D_{dispatch} is between 0 and 1, the larger γ_i is, the smaller d_i is assigned.

- A user inputs the overall minimum detection rate D_{target}.
- Select 100 pictures as training examples
  P=set of all text CCs in the training set
  N=set of all non-text CCs in the training set
- i=0; F_0=1; D_0=1; N_1=N;
- $Features$=$\{feature_j | j=1, \ldots, M\}$;

While $i<M$
  i=i+1;
  D_i=D_{i-1};
  For each $feature_j$ in $Features$
    Get distribution of $feature_j$ on $\{P, N_i\}$;
    Calculate $d_j(D_i)$, $f_j(D_i)$, $FR_j(D_i, 1-D_i)$;
  Choose $feature_k$ with the biggest $f_j(D_i)$;
  γ=$FR_k(D_i, 1-D_i)/SUM_j(FR_j(D_i, 1-D_i))$;
  d_i=$(D_{\text{target}}/D_i)\hat{}\gamma$;
  training: d_i=$h_i(d_i, P, N)$;
  Evaluate the current cascaded detector h_i on the set of non-text CCs and put any false detections into the set N_{i+1};
  D_i=$D_i \times d_i$;
  $Features$=$Features-feature_k$;

Fig.5 The algorithm for cascade training

In the training process, we retain as many text CCs as possible. Though there are many non-text CCs in the intermediate result, SVM in the following stage can filter them out.

REFINED CLASSIFICATION

The stage of precise classification is implemented by SVM to do further verification on the intermediate result of previous coarse classification. In our approach, with the help of cascade classifier, all 17 features are prepared at relatively small computational cost. All CCs which passed through the cascade are input into the SVM with normalized features, and only those accepted by SVM are considered as texts. Fig.6 shows classification result of SVM.

(a) (b)

Fig.6 Effect of SVM. (a) Input: CCs accepted by cascade; (b) Output: CCs accepted by SVM

We train SVM on a subset of training CC base instead of on the whole set, in order to make the SVM more suitable for 2-stage classification. We treat text CCs and non-text CCs differently. All text CCs are retained while we put non-text CCs into the cascade classifier. And those misclassified by the cascade as positive are merged with all text CCs to form the training set for SVM. Because we require the distribution of training CCs to be close to what the SVM will meet in practical application.

EXPERIMENTS

Taking into account that text CCs are usually large while false-accepted non-text CCs are small, it is unfair to directly compare the number of finally accepted text CCs and non-text CCs. Instead, we employ a strict pixel-wise evaluation method which compares output image with ground-truth image, as shown in Fig.7.

(a) (b) (c)

Fig.7 Evaluation details. (a) Original image; (b) Resulting binary image; (c) Ground truth image

In Fig.7, the resulting image and ground-truth image are binary images where text pixels are labelled as true in black. To ensure the soundness of evaluation, all the ground-truth images are labelled manually. The details of evaluation are shown in Eq.(7):

$$\left.\begin{aligned} hit &= area(Result \,\&\, GroundTruth),\\ error &= area(Result \,\&\, \overline{GroundTruth}),\\ miss &= area(\overline{Result} \,\&\, GroundTruth),\\ precision &= \frac{hit}{hit+error},\quad recall = \frac{hit}{hit+miss}, \end{aligned}\right\} \quad (7)$$

where *Result* is binary output image, *GroundTruth* is binary ground-truth image.

Our system is implemented on Pentium IV 3 GHz with VC++ 6.0. We build a testing base containing 500 scene images (640×480) in natural scenes with variant fonts, languages, skew angles and illumination conditions. The average processing time is less than 1 s and the performance detail is shown in Table 3. Some of our results are shown in Fig.8.

Table 3 Performance of the system

	Number of pictures	Precision (%)	Recall (%)
Training set	100	92.56	94.77
Test set	500	90.92	93.24

CONCLUSION

This paper presents a novel approach on scene text detection and segmentation. The method is learning-based and robust for various text size, font, language, color, skew angle and surface (Figs.8d~8f, 8j~8k). But for text on metal surface, because text is contaminated by specular reflectance, the result is unsatisfactory (Fig.8l).

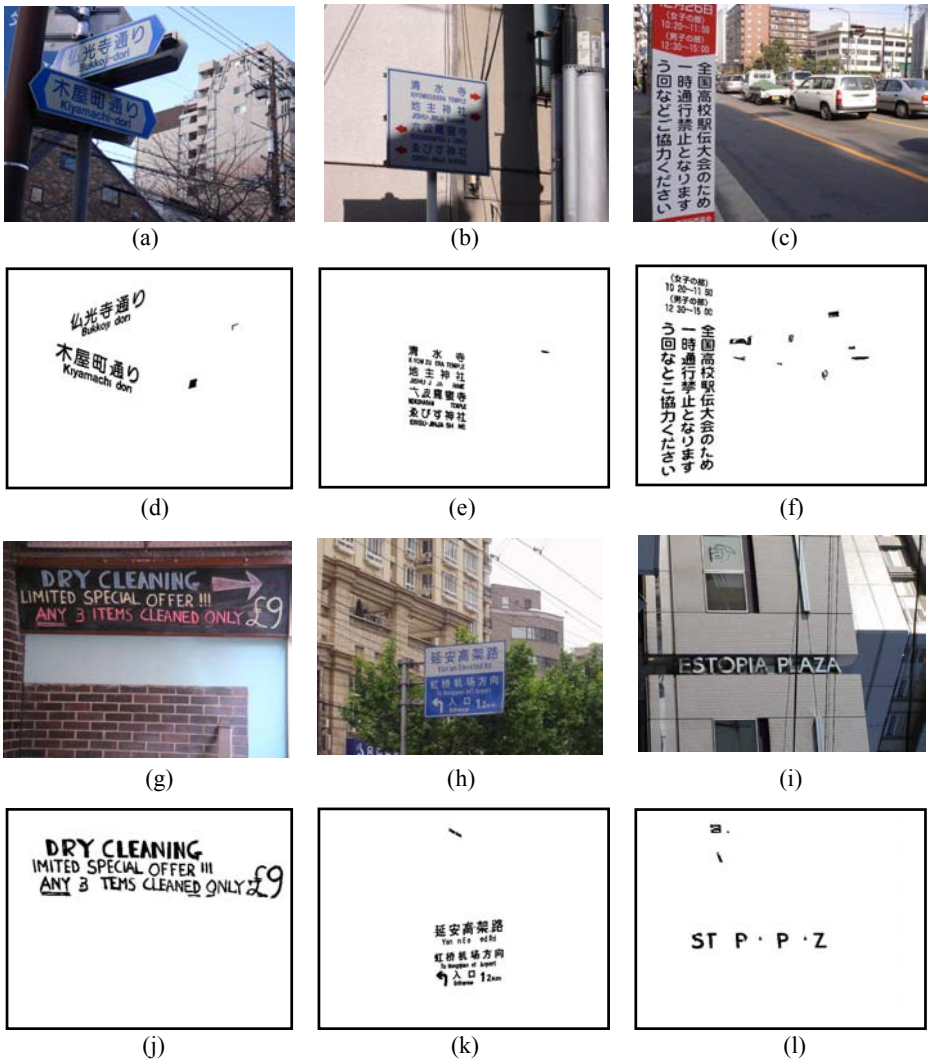


Fig.8 Some of our results. (a)~(c), (g)~(i) are original images; (d)~(f), (j)~(l) are result images

Further work will proceed on improvement of our approach. In current architecture, features are chosen by their error rate only, without consideration of computation cost. We will concentrate on combining features' effectiveness and efficiency in cascade training to speed up our algorithm.

References

Chen, D., Shearer, K., Bourlard, H., 2001. Text Enhancement with Symmetric Alter for Video OCR. Proc. International Conference on Image Analysis and Recognition, p.192-197.

Chun, B.T., Bae, Y., Kim, T.Y., 1999. Automatic Text Extraction in Digital Videos Using FFT and Neural Network. Proc. IEEE International Fuzzy Systems Conference. Seoul, Korea, **2**:1112-1115.

Clark, P., Mirmehdi, M., 2000. Finding Text Regions Using Localized Measures. Proc. 11th British Machine Vision Conference, p.675-684.

Ekin, A., 2006. Local Information Based Overlaid Text Detection by Classifier Fusion. Proc. International Conference on Acoustics, Speech and Signal Processing, **2**:753-756.

Kim, K.I., Jung, K., Kim, J.H., 2003. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Trans. Pattern Anal. Machine Intell.*, **25**(12):1631-1639. [doi:10.1109/TPAMI.2003.1251157]

Kim, K.C., Byun, H.R., Song, Y.J., Choi, Y.W., Chi, S.Y., Kim, K.K., Chung, Y.K., 2004. Scene Text Extraction in Natural Scene Images Using Hierarchical Feature Combining and Verification. Proc. International Conference on Computer Vision and Pattern Recognition, **2**:679-682.

Liu, C., Wang, C., Dai, R., 2005. Text Detection in Images Based on Unsupervised Classification of Edge-based Features. Proc. International Conference on Document Analysis and Recognition.

Liu, C.L., Koga, M., Fujisawa, H., 2005. Gabor Feature Extraction for Character Recognition: Comparison with Gradient Feature. Proc. 8th International Conference on Document Analysis and Recognition, **1**:121-125.

Lyu, M.R., Song, J., Cai, M., 2005. A comprehensive method for multilingual video text detection, localization, and extraction. *IEEE Trans. Circuits Syst. Video Technol.*, **15**(2):243-255. [doi:10.1109/TCSVT.2004.841653]

Mao, W., Chung, F., Lanm, K., Siu, W., 2002. Hybrid Chinese/English Text Detection in Images and Video Frames. Proc. International Conference on Computer Vision and Pattern Recognition, **3**:1015-1018.

Qian, X., Liu, G., 2006. Text Detection, Localization and Segmentation in Compressed Videos. Proc. International Conference on Acoustics, Speech and Signal Processing, **2**:385-388.

Takahashi, H., Nakajima, M., 2005. Region Graph Based Text Extraction from Outdoor Images. Proc. 3rd International Conference on Information Technology and Applications, **1**:680-685. [doi:10.1109/ICITA.2005.235]

Wang, K.Q., Kangas, J.A., 2003. Character location in scene images from digital camera. *Pattern Recognition*, **36**(10):2287-2299. [doi:10.1016/S0031-3203(03)00082-7]

Weinman, J., Hanson, A., McCallum, A., 2004. Sign Detection in Natural Images with Conditional Random Fields. Proc. IEEE International Workshop on Machine Learning for Signal Processing. Brazil, p.549-558. [doi:10.1109/MLSP.2004.1423018]

Winger, L., Robinson, J.A., Jernigan, M.E., 2000. Low-complexity character extraction in low-contrast scene images. *IEEE Trans. Pattern Recog. Artif. Intell.*, **14**(2):113-135. [doi:10.1142/S0218001400000106]

Zhang, D.Q., Chang, F.H., 2004. Learning to Detect Scene Text Using a Higher-Order MRF with Belief Propagation. Proc. International Conference on Computer Vision and Pattern Recognition, p.101-107.

Zhu, K., Qi, F., Jiang, R., Xu, L., 2005. Using Adaboost to Detect and Segment Characters from Natural Scenes. Proc. Conference on Camera Based Document Analysis and Recognition, p.52-59.